

## AppCompatActivity 類別

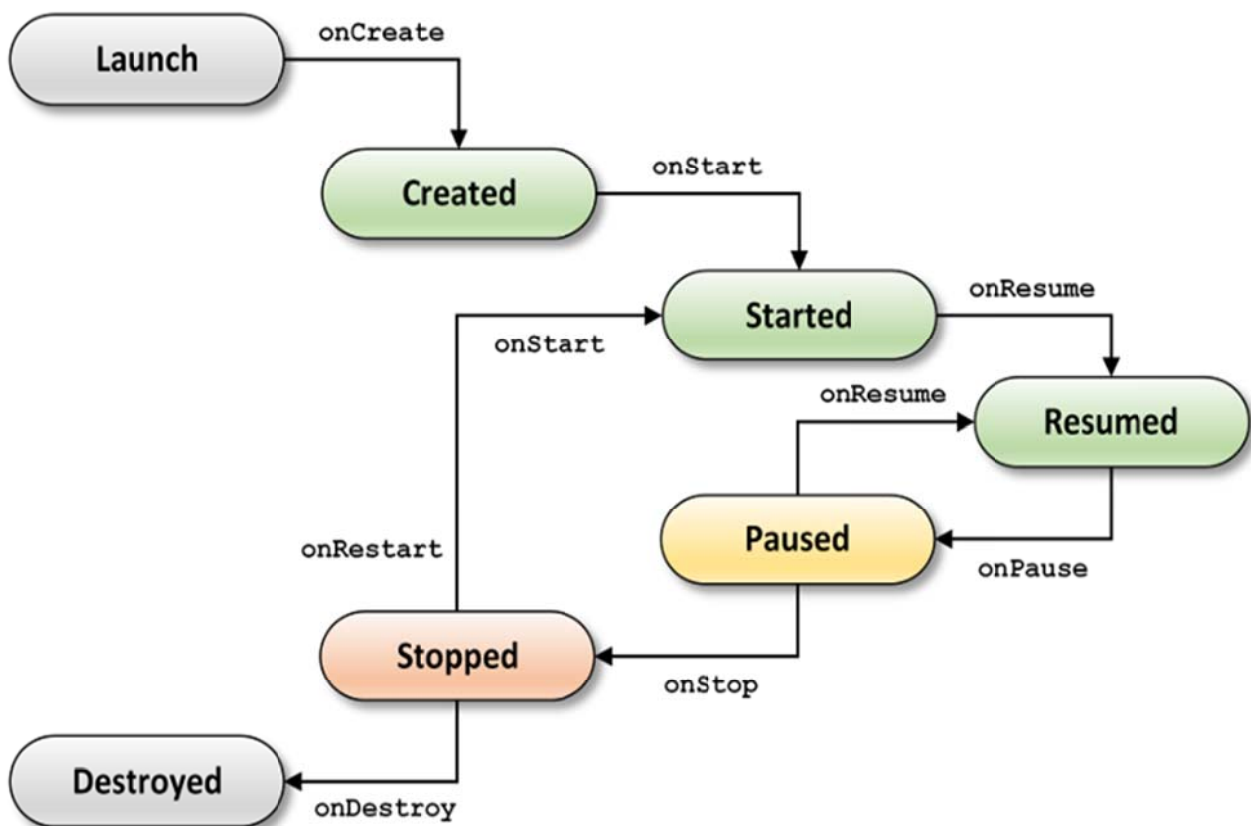
public class AppCompatActivity extends FragmentActivity implements  
AppCompatActivity, TaskStackBuilder.SupportParentable,  
ActionBarDrawerToggle.DelegateProvider

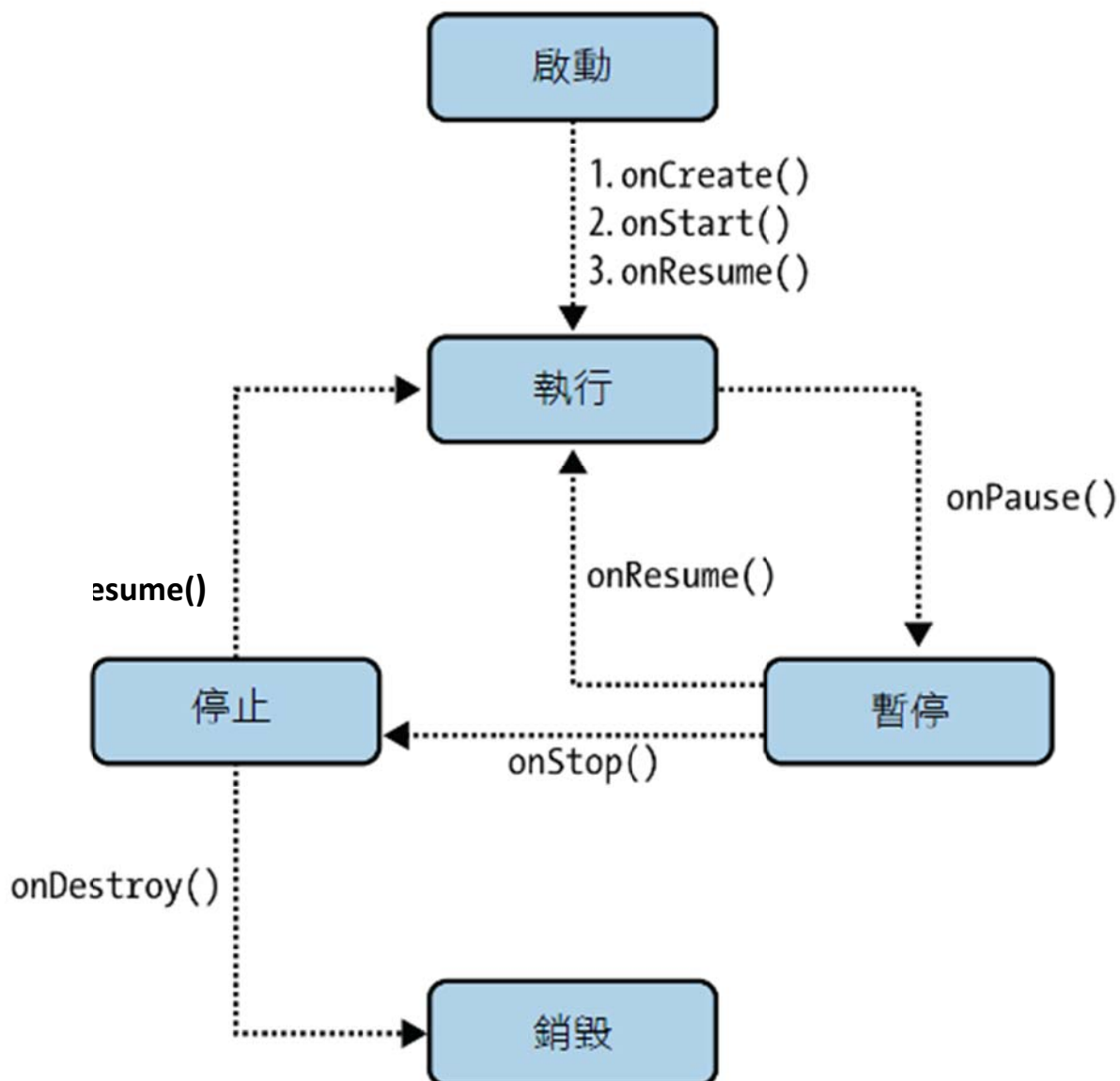
java.lang.Object

- ↳ android.content.Context
- ↳ android.content.ContextWrapper
- ↳ android.view.ContextThemeWrapper
- ↳ android.app.Activity
- ↳ android.support.v4.app.FragmentActivity
- ↳ android.support.v7.app.AppCompatActivity

註:Android Support Library

- ①用途是確保新版開發的應用程式可以在舊 age 版本的裝置上正常執行
- ②support.v4:向下支援到 Android 1.6
- ③support.v7:向下支援到 Android 2.1





方法:

Protected methods	
void	<code>onCreate(Bundle savedInstanceState)</code> Perform initialization of all fragments and loaders.
void	<code>onRestart()</code> Called after your activity has been stopped, prior to it being started again. Always followed by <code>onStart()</code>
void	<code>onStart()</code> Called when the activity is becoming visible to the user.
void	<code>onPostResume()</code> Dispatch <code>onResume()</code> to fragments.
void	<code>onResume()</code>

	Called when the activity will start interacting with the user. At this point your activity is at the top of the activity stack, with user input going to it.
void	<code>onPause()</code> Called when the system is about to start resuming a previous activity.
void	<code>onStop()</code> Called when the activity is no longer visible to the user, because another activity has been resumed and is covering this one.
void	<code>onDestroy()</code> The final call you receive before your activity is destroyed.
<b>Public methods</b>	
View	<code>findViewById(int id)</code> Finds a view that was identified by the id attribute from the XML that was processed in <code>onCreate(Bundle)</code> .
void	<code>finish()</code> Call this when your activity is done and should be closed.
Context	<code>getApplicationContext()</code> Return the context of the single, global Application object of the current process.
Intent	<code>getIntent()</code> Return the intent that started this activity.
LayoutInflater	<code>getLayoutInflater()</code> Convenience for calling <code>getLayoutInflater()</code> .
final void	<code>runOnUiThread(Runnable action)</code> Runs the specified action on the UI thread.
void	<code>setActionBar(toolbar)</code> Set a <code>Toolbar</code> to act as the <code>ActionBar</code> for this Activity window.
final void	<code>setResult(int resultCode, Intent data)</code> Call this to set the result that your activity will return to its caller.
void	<code>startActivity(Intent intent)</code> Same as <code>startActivity(Intent, Bundle)</code> with no options specified.
void	<code>startActivity(Intent intent, Bundle options)</code> Launch a new activity.
void	<code>startActivityForResult(Intent intent, int requestCode)</code> Same as calling <code>startActivityForResult(Intent, int, Bundle)</code> with no options
void	<code>startActivityForResult(Intent intent, int requestCode, Bundle options)</code> Launch an activity for which you would like a result when it finished.

## View 類別

public class View extends Object implements Drawable.Callback,  
KeyEvent.Callback, AccessibilityEventSource

java.lang.Object

↳ android.view.View

方法:

Public methods	
int	getId() Returns this view's identifier.
String	toString() Returns a string representation of the object.

## Log 類別

public final class Log extends Object

java.lang.Object

↳ android.util.Log

方法:

Public methods	
static int	d(String tag, String msg) Send a DEBUG log message.
static int	e(String tag, String msg) Send an ERROR log message.
static int	i(String tag, String msg) Send an INFO log message.

## Toast 類別

public class Toast extends Object

java.lang.Object

↳ android.widget.Toast

方法:

Public methods	
<code>static Toast</code>	<code>makeText(Context context, CharSequence text, int duration)</code> Make a standard toast that just contains a text view.
<code>void</code>	<code>setGravity(int gravity, int xOffset, int yOffset)</code> Set the location at which the notification should appear on the screen.
<code>void</code>	<code>setText(CharSequence s)</code> Update the text in a Toast that was previously created using one of the <code>makeText()</code> method.
<code>void</code>	<code>show()</code> Show the view for the specified duration.

說明:gravity 常數

<code>Gravity.CENTER</code>	水平、垂直都置中
<code>Gravity.CENTER_HORIZONTAL</code>	水平置中
<code>Gravity.CENTER_VERTICAL</code>	垂直置中
<code>Gravity.RIGHT</code>	靠螢幕右邊緣
<code>Gravity.LEFT</code>	靠螢幕左邊緣
<code>Gravity.TOP</code>	靠螢幕上邊緣
<code>Gravity.BOTTOM</code>	靠螢幕下邊緣

## Button 類別

`public class Button extends TextView`

`java.lang.Object`

↳ `android.view.View`

↳ `android.widget.TextView`

↳ `android.widget.Button`

方法:

Constructors	
<code>Button(Context context)</code>	
Public methods	
<code>CharSequence</code>	<code>getAccessibilityClassName()</code> Return the class name of this object to be used for accessibility purposes.

## View.OnClickListener 介面

public static interface View.OnClickListener

android.view.View.OnClickListener

方法:

Public methods	
abstract void	onClick(View v) Called when a view has been clicke

範例:按鈕 onClick 事件處理

方式 1:

①xml 中定義 onClick 事件方法

**android:onClick="click1"**

②撰寫 click1 方法

```
public void click1(View v)
```

//無須載入 Button 元件

方式 2:Anonymous inner class

```
Button btn=(Button) findViewById(R.id.xxx);
```

```
btn.setOnClickListener(new View.OnClickListener() {
```

```
    @Override
```

```
    public void onClick(View v) {
```

```
        Toast.makeText(MainActivity.this,"hi",Toast.LENGTH_SHORT).show();
```

```
    }
```

```
});
```

方式 3:Activity implement View.OnClickListener

```
public class MainActivity extends AppCompatActivity
```

```
    implements View.OnClickListener {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        Button btn=(Button) findViewById(R.id.button);
```

```
        btn.setOnClickListener(getApplicationContext());
```

```
    }
```

```

@Override
public void onClick(View v) {
    Toast.makeText(getBaseContext(),"hi",Toast.LENGTH_SHORT).show();
}
}

```

方式 4:自訂 Member inner class implement View.OnClickListener

```

public class MainActivity extends AppCompatActivity {
    private Button btn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btn=(Button) findViewById(R.id.button);
        btn.setOnClickListener(new MyListener());
    }
    class MyListener implements View.OnClickListener {
        @Override
        public void onClick(View v) {
            Toast.makeText(this,"hi",Toast.LENGTH_SHORT).show();
        }
    }
}

```

例.按鈕共用 Listener

```

public class MainActivity extends AppCompatActivity {
    private TextView tv;
    private Button btn,btn2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        tv=(TextView) findViewById(R.id.textView);
        btn=(Button) findViewById(R.id.button);
        btn2=(Button) findViewById(R.id.button2);
        btn.setOnClickListener(myListener);
        btn2.setOnClickListener(myListener);
    }
}

```

```

private Button.OnClickListener myListener=new Button.OnClickListener() {
    @Override
    public void onClick(View v) {

        switch (v.getId()) {
            case R.id.button:
                tv.setText("Button A");
                break;
            case R.id.button2:
                tv.setText("Button B");
                break;
        }
    }
};
}

```

註: **Button.OnClickListener** 可以改用 **View.OnClickListener**

例.

```

private Button.OnClickListener myListener=new Button.OnClickListener()
{
    @Override
    public void onClick(View v) {
        switch (v.getId())
        {
            case R.id.button:
                Toast.makeText(MainActivity.this,"default",Toast.LENGTH_SHORT).show();
                break;
            case R.id.button2:
                Toast toast=Toast.makeText(MainActivity.this,
                    "Gravity.CENTER+(0,-450)",Toast.LENGTH_SHORT);
                toast.setGravity(Gravity.CENTER,0,-450);
                toast.show();
                break;
        }
    }
};
}

```

註:Activity 類別.this(即 Context)可以使用 **getBaseContext()** 取得



## TextView 類別

public class TextView extends View

implements ViewTreeObserver.OnPreDrawListener

java.lang.Object

↳ android.view.View

↳ android.widget.TextView

方法:

Public methods	
CharSequence	<a href="#">getText()</a> Return the text the TextView is displaying.
final void	<a href="#">setHint(int resid)</a> Sets the text to be displayed when the text of the TextView is empty, from a resource.
final void	<a href="#">setText(CharSequence text)</a> Sets the string value of the TextView.
void	<a href="#">setTextColor(int color)</a> Sets the text color for all the states (normal, selected, focused) to be this color.
void	<a href="#">setTextSize(float size)</a> Set the default text size to the given value, interpreted as "scaled pixel" units.

## EditText 類別

public class EditText extends TextView

ava.lang.Object

↳ android.view.View

↳ android.widget.TextView

↳ android.widget.EditText

方法:

Public methods	
Editable	<a href="#">getText()</a> Return the text the TextView is displaying.
void	<a href="#">selectAll()</a> Convenience for <a href="#">selectAll(Spannable)</a> .

## AlertDialog.Builder 類別

public static class AlertDialog.Builder extends Object

java.lang.Object

↳ android.app.AlertDialog.Builder

方法:

Public constructors	
AlertDialog.Builder(Context context)	Creates a builder for an alert dialog that uses the default alert dialog theme.
Public methods	
AlertDialog.Builder.setIcon(int iconId)	Set the resource id of the Drawable to be used in the title.
AlertDialog.Builder.setMessage(CharSequence message)	Set the message to display.
AlertDialog.Builder.setText(CharSequence s)	Update the text in a Toast that was previously created using one of the makeText() method.
AlertDialog.Builder.setNegativeButton(CharSequence text, DialogInterface.OnClickListener listener)	Set a listener to be invoked when the negative button of the dialog is pressed.
AlertDialog.Builder.setNeutralButton(CharSequence text, DialogInterface.OnClickListener listener)	Set a listener to be invoked when the neutral button of the dialog is pressed.
AlertDialog.Builder.setPositiveButton(CharSequence text, DialogInterface.OnClickListener listener)	Set a listener to be invoked when the positive button of the dialog is pressed.
AlertDialog.Builder.setTitle(CharSequence title)	Set the title displayed in the Dialog.
AlertDialog.show()	Creates an AlertDialog with the arguments supplied to this builder and immediately displays the dialog.

## DialogInterface.OnClickListener 介面

public static interface DialogInterface.OnClickListener

android.content.DialogInterface.OnClickListener

方法:

Public methods	
<code>abstract void</code>	<code>onClick(DialogInterface dialog, int which)</code>
This method will be invoked when a button in the dialog is clicked.	

例.

```
AlertDialog.Builder alert=new AlertDialog.Builder(MainActivity.this);
alert.setTitle("訊息對話框");
alert.setMessage("確定/取消");
alert.setPositiveButton("確定", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        :
    }
});
alert.setNegativeButton("取消", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        :
    }
});
alert.show();
```

## ImageButton 類別

```
public class ImageButton extends ImageView
```

```
java.lang.Object
```

```
↳ android.view.View
```

```
↳ android.widget.ImageView
```

```
↳ android.widget.ImageButton
```

方法:

Public constructors	
<code>ImageButton(Context context)</code>	
Public methods	
<code>CharSequence</code>	<code>getAccessibilityClassName()</code>
Return the class name of this object to be used for accessibility purposes.	

<code>void</code>	<code>setImageResource(int resId)</code> Sets a drawable as the content of this ImageView.
-------------------	---

例.翻牌

```
private ImageButton imageButton;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    imageButton=(ImageButton) findViewById(R.id.imageButton);
    imageButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            imageButton.setImageResource(R.drawable.hj);
        }
    });
}
```

## ToggleButton 類別

```
public class ToggleButton extends CompoundButton
```

```
java.lang.Object
```

```
↳ android.view.View
```

```
↳ android.widget.TextView
```

```
↳ android.widget.Button
```

```
↳ android.widget.CompoundButton
```

```
↳ android.widget.ToggleButton
```

方法:

Public constructors	
	<code>ToggleButton(Context context)</code>
Public methods	
<code>CharSequence</code>	<code>getTextOff()</code> Returns the text for when the button is not in the checked state.
<code>CharSequence</code>	<code>getTextOn()</code> Returns the text for when the button is in the checked state.
<code>void</code>	<code>setChecked(boolean checked)</code> Changes the checked state of this button.

<code>void</code>	<code>setTextOff(CharSequence textOff)</code> Sets the text for when the button is not in the checked state.
<code>void</code>	<code>setTextOn(CharSequence textOn)</code> Sets the text for when the button is in the checked state.

## CompoundButton 類別

```
public abstract class CompoundButton extends Button
                                   implements Checkable
```

java.lang.Object

↳ android.view.View

↳ android.widget.TextView

↳ android.widget.Button

↳ android.widget.CompoundButton

方法:

Public methods	
<code>void</code>	<code>setOnCheckedChangeListener(CompoundButton.OnCheckedChangeListener listener)</code> Register a callback to be invoked when the checked state of this button changes.

## CompoundButton.OnCheckedChangeListener 介面

```
public static interface CompoundButton.OnCheckedChangeListener
```

方法:

Public methods	
<code>abstract void</code>	<code>onCheckedChanged(CompoundButton buttonView, boolean isChecked)</code> Called when the checked state of a compound button has changed.

例.切换螢幕方向

```
public class MainActivity extends AppCompatActivity implements
                                   CompoundButton.OnCheckedChangeListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

```

ToggleButton toggleButton=(ToggleButton) findViewById(R.id.toggleButton);
toggleButton.setOnCheckedChangeListener(this);
}

public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
    if(isChecked)
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
    else
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
}
}

```

或

```

final ToggleButton toggleButton=(ToggleButton) findViewById(R.id.toggleButton);
toggleButton.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if(isChecked)
            setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE); //直
        else
            setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT); //橫
    }
});

```

## CheckBox 類別

```
public class CheckBox extends CompoundButton
```

```
java.lang.Object
```

```
↳ android.view.View
```

```
↳ android.widget.TextView
```

```
↳ android.widget.Button
```

```
↳ android.widget.CompoundButton
```

```
↳ android.widget.CheckBox
```

Direct Subclasses AppCompatActivity

方法:

Public methods	
boolean	<code>isChecked()</code>
void	<code>setChecked(boolean checked)</code> Changes the checked state of this button.
void	<code>setOnCheckedChangeListener(CompoundButton.OnCheckedChangeListener listener)</code> Register a callback to be invoked when the checked state of this button changes.
CharSequence	<code>getText()</code> Return the button title.
void	<code>setText(String text)</code> Sets the string value of the button.

## RadioButton 類別

```
public class RadioButton extends CompoundButton
```

```
java.lang.Object
```

```
↳ android.view.View
```

```
↳ android.widget.TextView
```

```
↳ android.widget.Button
```

```
↳ android.widget.CompoundButton
```

```
↳ android.widget.RadioButton
```

方法:

Public methods	
boolean	<code>isChecked()</code>
void	<code>setChecked(boolean checked)</code> Changes the checked state of this button.
void	<code>setOnCheckedChangeListener(CompoundButton.OnCheckedChangeListener listener)</code> Register a callback to be invoked when the checked state of this button changes.
CharSequence	<code>getText()</code> Return the button title.
void	<code>setText(String text)</code> Sets the string value of the button.

## RadioGroup 類別

`public class RadioGroup extends LinearLayout`

`java.lang.Object`

```
↳ android.view.View
    ↳ android.view.ViewGroup
        ↳ android.widget.LinearLayout
            ↳ android.widget.RadioGroup
```

方法:

Public methods	
<code>int</code>	<code>getCheckedRadioButtonId()</code> Returns the identifier of the selected radio button in this group.
<code>void</code>	<code>setOnCheckedChangeListener(RadioGroup.OnCheckedChangeListener listener)</code> Register a callback to be invoked when the checked radio button changes in this group.

## RadioGroup.OnCheckedChangeListener 介面

`public static interface RadioGroup.OnCheckedChangeListener`

方法:

Public methods	
<code>abstract void</code>	<code>onCheckedChanged(RadioGroup group, int checkedId)</code> Called when the checked radio button has changed.

例.

```
RadioGroup rg;
:
rg.setOnCheckedChangeListener(myListener);
:
RadioGroup.OnCheckedChangeListener myListener=new
                                RadioGroup.OnCheckedChangeListener() {
    public void onCheckedChanged(RadioGroup group, int checkedId) {
        :
    }
};
```



## ImageView 類別

`public class ImageView extends View`

`java.lang.Object`

↳ `android.view.View`

↳ `android.widget.ImageView`

方法:

Public methods	
<code>ImageView.ScaleType</code>	<code>getScaleType()</code> Return the current scale type in use by this ImageView.
<code>void</code>	<code>setImageResource(int resId)</code> Sets a drawable as the content of this ImageView.
<code>void</code>	<code>setLayoutParams (GridView.LayoutParams)</code> Set the layout parameters associated with this view.

## ImageView.ScaleType 列舉

Enum values	
<code>ImageView.ScaleType</code>	<code>CENTER</code> Center the image in the view, but perform no scaling.
<code>ImageView.ScaleType</code>	<code>CENTER_CROP</code> Scale the image uniformly (maintain the image's aspect ratio) so that both dimensions (width and height) of the image will be equal to or larger than the corresponding dimension of the view (minus padding).
<code>ImageView.ScaleType</code>	<code>CENTER_INSIDE</code> Scale the image uniformly (maintain the image's aspect ratio) so that both dimensions (width and height) of the image will be equal to or less than the corresponding dimension of the view (minus padding).
<code>ImageView.ScaleType</code>	<code>FIT_CENTER</code> Scale the image using <code>CENTER</code> .
<code>ImageView.ScaleType</code>	<code>FIT_END</code> Scale the image using <code>END</code> .
<code>ImageView.ScaleType</code>	<code>FIT_START</code> Scale the image using <code>START</code> .
<code>ImageView.ScaleType</code>	<code>FIT_XY</code> Scale the image using <code>FILL</code> .
<code>ImageView.ScaleType</code>	<code>MATRIX</code> Scale using the image matrix when drawing.

## GridView 類別

```
public class GridView extends AbsListView
```

```
java.lang.Object
```

```
↳ android.view.View
```

```
↳ android.view.ViewGroup
```

```
↳ android.widget.AdapterView<android.widget.ListAdapter>
```

```
↳ android.widget.AbsListView
```

```
↳ android.widget.GridView
```

XML	
<code>android:columnWidth</code>	Specifies the fixed width for each column.
<code>android:gravity</code>	Specifies the gravity within each cell.
<code>android:horizontalSpacing</code>	Defines the default horizontal spacing between columns.
<code>android:numColumns</code>	Defines how many columns to show.
<code>android:stretchMode</code>	Defines how columns should stretch to fill the available empty space, if any.
<code>android:verticalSpacing</code>	Defines the default vertical spacing between rows.

方法:

Public methods	
<code>void</code>	<code>setAdapter(ListAdapter adapter)</code> Sets the data behind this GridView.
<code>void</code>	<code>setOnItemClickListener()</code> new AdapterView.OnItemClickListener() override onItemClick method.

## GridLayout.LayoutParams 類別

```
public static class GridLayout.LayoutParams
```

```
extends ViewGroup.MarginLayoutParams
```

方法:

Public constructors	
<code>GridLayout.LayoutParams</code>	<code>(GridLayout.Spec rowSpec, GridLayout.Spec columnSpec)</code>

## Spinner 類別

```
public class Spinner extends AbsSpinner
                        implements DialogInterface.OnClickListener
```

```
java.lang.Object
```

```
↳ android.view.View
    ↳ android.view.ViewGroup
        ↳ android.widget.AdapterView<android.widget.SpinnerAdapter>
            ↳ android.widget.AbsSpinner
                ↳ android.widget.Spinner
```

例.

```
Spinner spinner = (Spinner) findViewById(R.id.spinner);
spinner.setOnItemSelectedListener(this);
```

①資料必須由 Adapter 取得

②資料來源可以是 ❶strings.xml ❷array ❸List ❹資料庫

## AdapterView 類別

```
public abstract class AdapterView extends ViewGroup
```

```
java.lang.Object
```

```
↳ android.view.View
    ↳ android.view.ViewGroup
        ↳ android.widget.AdapterView<T extends android.widget.Adapter>
```

Direct Subclasses:

AbsListView, AbsSpinner, AdapterViewAnimator

Indirect Subclasses:

AdapterViewFlipper, AppCompatActivity, ExpandableListView, Gallery,  
GridView, ListView, **Spinner**, StackView

方法:

Public methods	
int	getSelectedItemPosition() Return the position of the currently selected item within the adapter's data set
Object	getSelectedItem()
long	getSelectedItemId()

例.

```
int position = spinner.getSelectedItemPosition();
```

## ArrayAdapter 類別

```
public class ArrayAdapter extends BaseAdapter
    implements Filterable, ThemedSpinnerAdapter
```

```
java.lang.Object
```

```
↳ android.widget.BaseAdapter
```

```
↳ android.widget.ArrayAdapter<T>
```

方法:

Public constructors	
<code>ArrayAdapter(Context context, int resource, T[] objects)</code>	
<code>ArrayAdapter(Context context, int resource, List&lt;T&gt; objects)</code>	
Public methods	
<code>Static ArrayAdapter&lt;CharSequence&gt;</code>	<code>createFromResource(Context context, int textArrayResId, int textViewResId)</code> Creates a new ArrayAdapter from external resources.
<code>int</code>	<code>getCount()</code> How many items are in the data set represented by this Adapter.
<code>int</code>	<code>getPosition(T item)</code> Returns the position of the specified item in the array.

例.

```
ArrayAdapter<String> adp=
    new ArrayAdapter<>(this,android.R.layout.simple_spinner_item,dessert);
```

```
spinner.setAdapter(adp);
```

註:資料來源是 array/ List 建立

例.

```
ArrayAdapter<CharSequence> adp=
    ArrayAdapter.createFromResource(MainActivity.this,R.array.dessert,
        android.R.layout.simple_spinner_item);
```

```
spinner.setAdapter(adp);
```

註:資料來源是 strings.xml 定義

## SimpleAdapter 類別

```
public class SimpleAdapter extends BaseAdapter
    implements Filterable, ThemedSpinnerAdapter
```

java.lang.Object

↳ android.widget.BaseAdapter

↳ android.widget.SimpleAdapter

## AppCompatActivity 類別

```
public class AppCompatActivity extends Spinner
    implements TintableBackgroundView
```

android.widget.Spinner

↳ android.support.v7.widget.AppCompatActivity

## AdapterView.OnItemSelectedListener 介面

```
public static interface AdapterView.OnItemSelectedListener
```

android.widget.AdapterView.OnItemSelectedListener

方法:

Public methods	
abstract void	onItemSelected(AdapterView<?> parent, View view, int position, long id) Callback method to be invoked when an item in this view has been selected.
abstract void	onNothingSelected(AdapterView<?> parent) Callback method to be invoked when the selection disappears from this view.

## ListView 類別

```
public class ListView extends AbsListView
```

java.lang.Object

↳ android.view.View

↳ android.view.ViewGroup

↳ android.widget.AdapterView<android.widget.ListAdapter>

↳ android.widget.AbsListView

↳ android.widget.ListView

## BaseAdapter 類別

```
public abstract class BaseAdapter extends Object
                                implements ListAdapter, SpinnerAdapter
```

java.lang.Object

↳ android.widget.BaseAdapter

## Adapter 介面

```
public interface Adapter
```

android.widget.Adapter

### Indirect Subclasses

ArrayAdapter<T>, **BaseAdapter**, CursorAdapter, HeaderViewListAdapter, ListAdapter, ResourceCursorAdapter, SimpleAdapter, SimpleCursorAdapter, SpinnerAdapter, ThemedSpinnerAdapter, WrapperListAdapter

方法:

Public methods	
abstract void	<pre>getCount()</pre> <p>How many items are in the data set represented by this Adapter.</p>
abstract void	<pre>getItem(int position)</pre> <p>Get the data item associated with the specified position in the data set.</p>
abstract long	<pre>getItemId(int position)</pre> <p>Get the row id associated with the specified position in the list</p>
abstract View	<pre>getView(int position, View convertView, ViewGroup parent)</pre> <p>Get a View that displays the data at the specified position in the data set</p>

## Intent 類別

```
public class Intent extends Object implements Parcelable, Cloneable
```

java.lang.Object

↳ android.content.Intent

方法:

Public constructors
<pre>Intent()</pre>

Create an empty intent.	
<code>Intent(Context packageContext, Class&lt;?&gt; cls)</code>	
Create an intent for a specific component.	
<b>Public methods</b>	
<code>Intent</code>	<code>putExtra(String name, int value)</code> Add extended data to the intent.
<code>Intent</code>	<code>putExtras(Bundle extras)</code> Add a set of extended data to the intent.
<code>static Intent</code>	<code>getIntent()</code> <i>This method was deprecated in API level 4. Use <code>parseUri(String, int)</code> instead</i>
<code>int</code>	<code>getIntExtra(String name, int defaultValue)</code> Retrieve extended data from the intent.
<code>long</code>	<code>getLongExtra(String name, long defaultValue)</code> Retrieve extended data from the intent.
<code>String</code>	<code>getStringExtra(String name)</code> Retrieve extended data from the intent.
<code>float</code>	<code>getFloatExtra(String name, float defaultValue)</code> Retrieve extended data from the intent.
<code>double</code>	<code>getDoubleExtra(String name, double defaultValue)</code> Retrieve extended data from the intent.
<code>ArrayList&lt;Integer&gt;</code>	<code>getIntegerArrayListExtra(String name)</code> Retrieve extended data from the intent.

例.啟動另一個 Activity

```
Intent intent=new Intent(MainActivity.this, SecondActivity.class);
startActivity(intent);
```

例.啟動另一個 Activity, 同時傳遞參數

```
Intent intent =new Intent(this,SecondActivity.class);
String str=editText.getText().toString();
intent.putExtra("myData",str);
startActivity(intent);
```

## Bundle 類別

```
public final class Bundle extends BaseBundle implements Cloneable, Parcelable
java.lang.Object
```

↳ android.os.BaseBundle

↳ android.os.Bundle

方法:

Public constructors	
<code>Bundle()</code>	
Constructs a new, empty Bundle.	
Public methods	
<code>boolean</code>	<code>getBoolean(String key, boolean defaultValue)</code> Returns the value associated with the given key, or defaultValue if no mapping of the desired type exists for the given key.
<code>double</code>	<code>getDouble(String key, double defaultValue)</code> Returns the value associated with the given key, or defaultValue if no mapping of the desired type exists for the given key.
<code>int</code>	<code>getInt(String key, int defaultValue)</code> Returns the value associated with the given key, or defaultValue if no mapping of the desired type exists for the given key.
<code>long</code>	<code>getLong(String key, long defaultValue)</code> Returns the value associated with the given key, or defaultValue if no mapping of the desired type exists for the given key.
<code>String</code>	<code>getString(String key, String defaultValue)</code> Returns the value associated with the given key, or defaultValue if no mapping of the desired type exists for the given key or if a null value is explicitly associated with the given key.
<code>void</code>	<code>putBoolean(String key, boolean value)</code> Inserts a Boolean value into the mapping of this Bundle, replacing any existing value for the given key.
<code>void</code>	<code>putDouble(String key, double value)</code> Inserts a double value into the mapping of this Bundle, replacing any existing value for the given key.
<code>void</code>	<code>putInt(String key, int value)</code> Inserts an int value into the mapping of this Bundle, replacing any existing value for the given key.
<code>void</code>	<code>putLong(String key, long value)</code> Inserts a long value into the mapping of this Bundle, replacing any existing value for the given key.
<code>void</code>	<code>putString(String key, String value)</code> Inserts a String value into the mapping of this Bundle, replacing any existing value for the given key.



例.以 Bundle 打包參數

```
Bundle bundle=new Bundle();
String str="kuomulin";
bundle.putString("name",str);
bundle.putString("occupation","teacher");
Intent intent=new Intent(MainActivity.this,SecondActivity.class);
intent.putExtra(bundle);
startActivity(intent);
```

### DefaultHandler 類別

public class DefaultHandler extends Object implements  
EntityResolver, DTDHandler, ContentHandler, ErrorHandler

java.lang.Object

↳ org.xml.sax.helpers.DefaultHandler

方法:

Public methods	
void	characters(char[] ch, int start, int length) Receive notification of character data inside an element.
void	endElement(String uri, String localName, String qName) Receive notification of the end of an element.
void	startElement(String uri, String localName, String qName, Attributes attributes) Receive notification of the start of an element.

### LayoutInflater 類別

public abstract class LayoutInflater extends Object

java.lang.Object

↳ android.view.LayoutInflater

方法:

Public methods	
static LayoutInflater	from(Context context) Obtains the LayoutInflater from the given context.
View	inflate(int resource, ViewGroup root)

	Inflate a new view hierarchy from the specified xml resource.
--	---

例.

```
LayoutInflater inflater= LayoutInflater.from(MainActivity.this);
View v=inflater.inflate(R.layout.myitem,null);
```

## Handler 類別

public class Handler extends Object

java.lang.Object

↳ android.os.Handler

方法:

Public constructors	
Handler( )	
Public methods	
void	handleMessage(Message msg) Subclasses must implement this to receive messages.
final Message	obtainMessage(int what, Object obj) Same as obtainMessage(), except that it also sets the what and obj members of the returned Message.
final Message	obtainMessage() Returns a new Message from the global message pool.
final Message	obtainMessage(int what) Same as obtainMessage(), except that it also sets the what member of the returned Message
final boolean	post(Runnable r) Causes the Runnable r to be added to the message queue.
final boolean	postDelayed(Runnable r, long delayMillis) Causes the Runnable r to be added to the message queue, to be run after the specified amount of time elapses.
final boolean	sendMessage(Message msg) Pushes a message onto the end of the message queue after all pending messages before the current time.
final boolean	sendMessageDelayed(Message msg, long delayMillis) Enqueue a message into the message queue after all pending messages before (current time + delayMillis).

## Message 類別

`public final class Message extends Object implements Parcelable`

`java.lang.Object`

↳ `android.os.Message`

方法:

Public constructors	
<code>Message( )</code>	
Public fields	
<code>int</code>	<p><code>what</code></p> <p>User-defined message code so that the recipient can identify what this message is about.</p>
Public methods	
<code>static Message</code>	<p><code>obtain(Handler h)</code></p> <p>Same as <code>obtain()</code>, but sets the value for the <i>target</i> member on the Message returned.</p>
<code>Bundle</code>	<p><code>getData()</code></p> <p>Obtains a Bundle of arbitrary data associated with this event, lazily creating it if necessary.</p>
<code>void</code>	<p><code>setData(Bundle data)</code></p> <p>Sets a Bundle of arbitrary data values</p>

## AnimationDrawable 類別

`public class AnimationDrawable extends DrawableContainer`  
`implements Runnable, Animatable`

`java.lang.Object`

↳ `android.graphics.drawable.Drawable`

↳ `android.graphics.drawable.DrawableContainer`

↳ `android.graphics.drawable.AnimationDrawable`

方法:

Public methods	
<code>void</code>	<p><code>start()</code></p> <p>Starts the animation from the first frame, looping if necessary.</p>
<code>void</code>	<p><code>stop()</code></p> <p>Stops the animation at the current frame.</p>

**ContextCompat 類別**

```
public class ContextCompat extends Object
```

```
java.lang.Object
```

```
↳ android.support.v4.content.ContextCompat
```

方法:

Public methods	
<code>static final Drawable</code>	<code>getDrawable(Context context, int id)</code> Returns a drawable object associated with a particular resource ID.
<code>static void</code>	<code>startActivity(Context context, Intent intent, Bundle options)</code> Start an activity with additional launch information, if able.

**ProgressDialog 類別**

```
public class ProgressDialog extends AlertDialog
```

```
java.lang.Object
```

```
↳ android.app.Dialog
```

```
↳ android.app.AlertDialog
```

```
↳ android.app.ProgressDialog
```

方法:

Public constructors	
<code>ArrayAdapter(Context context, int resource, T[] objects)</code>	
Public methods	
<code>void</code>	<code>setProgressStyle(int style)</code> ①STYLE_HORIZONTAL: Creates a ProgressDialog with a horizontal progress bar. ②STYLE_SPINNER: Creates a ProgressDialog with a circular, spinning progress bar.(預設)
<code>void</code>	<code>endElement(String uri, String localName, String qName)</code> Receive notification of the end of an element.
<code>void</code>	<code>startElement(String uri, String localName, String qName, Attributes attributes)</code> Receive notification of the start of an element.

**URL 類別**

```
public final class URL extends Object implements Serializable
```

java.lang.Object

↳ java.net.URL

方法:

Public constructors	
<code>URL(String spec)</code>	
Creates a <code>URL</code> object from the <code>String</code> representation.	
Public methods	
<code>URLConnection</code>	<code>openConnection()</code> Returns a <code>URLConnection</code> instance that represents a connection to the remote object referred to by the <code>URL</code> .

## HttpURLConnection 類別

`public abstract class HttpURLConnection extends URLConnection`

java.lang.Object

↳ java.net.URLConnection

↳ java.net.HttpURLConnection

方法:

Public constructors	
<code>HttpURLConnection(URL u)</code> Constructor for the <code>HttpURLConnection</code> .	
Public methods	
<code>void</code>	<code>setRequestMethod(String method)</code> Set the method for the URL request, one of: <ul style="list-style-type: none"> <li>• GET</li> <li>• POST</li> </ul>
<code>abstract void</code>	<code>connect()</code> Opens a communications link to the resource referenced by this <code>URL</code> , if such a connection has not already been established.
<code>abstract void</code>	<code>disconnect()</code> Indicates that other requests to the server are unlikely in the near future.
<code>InputStream</code>	<code>getInputStream()</code> Returns an input stream that reads from this open connection.

## InputStream 類別

```
public abstract class InputStream
                        extends Object implements Closeable
```

java.lang.Object

↳ java.io.InputStream

Direct Subclasses

AssetManager.AssetInputStream, BackupDataInputStream, **ByteArrayInputStream**, FileInputStream, FilterInputStream, ObjectInputStream, PipedInputStream, SequenceInputStream, StringBufferInputStream

方法:

Public constructors	
<a href="#">InputStream()</a>	
Public methods	
int	<a href="#">available()</a> Returns an estimate of the number of bytes that can be read (or skipped over) from this input stream without blocking by the next invocation of a method for this input stream.
void	<a href="#">close()</a> Closes this input stream and releases any system resources associated with the stream.
int	<a href="#">read(byte[] b)</a> Reads some number of bytes from the input stream and stores them into the buffer array <b>b</b> .

## OutputStream 類別

```
public abstract class OutputStream extends Object
                        implements Closeable, Flushable
```

java.lang.Object

↳ java.io.OutputStream

Known Direct Subclasses

**ByteArrayOutputStream**, FileOutputStream, FilterOutputStream, ObjectOutputStream, PipedOutputStream

方法:

Public constructors	
<a href="#">OutputStream()</a>	

Public methods	
void	<b>close()</b> Closes this output stream and releases any system resources associated with this stream.
void	<b>flush()</b> Flushes this output stream and forces any buffered output bytes to be written out.
void	<b>write(byte[] b)</b> Writes <b>b.length</b> bytes from the specified byte array to this output stream.

## ByteArrayOutputStream 類別

**public class ByteArrayOutputStream extends OutputStream**

java.lang.Object

↳ java.io.OutputStream

↳ java.io.ByteArrayOutputStream

方法:

Public constructors	
	<b>ByteArrayOutputStream()</b> Creates a new byte array output stream.
Public methods	
void	<b>close()</b> Closing a ByteArrayOutputStream has no effect.
void	<b>write(byte[] b, int off, int len)</b> Writes <b>len</b> bytes from the specified byte array starting at offset <b>off</b> to this byte array output stream.

## XMLReader 介面

**public interface XMLReader**

org.xml.sax.XMLReader

方法:

Public methods	
abstract void	<b>parse(InputSource input)</b> Parse an XML document.
abstract void	<b>parse(String systemId)</b> Parse an XML document from a system identifier (URI).

<b>abstract void</b>	<code>setContentHandler(ContentHandler handler)</code> Allow an application to register a content event handler.
----------------------	---

## SAXParser 類別

`public abstract class SAXParser extends Object`

`java.lang.Object`

↳ `javax.xml.parsers.SAXParser`

方法:

Public methods	
<b>abstract XMLReader</b>	<code>getXMLReader()</code> Returns the <code>XMLReader</code> that is encapsulated by the implementation of this class.

## SAXParserFactory 類別

`public abstract class SAXParserFactory extends Object`

`java.lang.Object`

↳ `javax.xml.parsers.SAXParserFactory`

方法:

Public methods	
<b>static SAXParserFactory</b>	<code>newInstance(String factoryClassName, ClassLoader classLoader)</code> Returns an instance of the named implementation of <code>SAXParserFactory</code> .
<b>static SAXParserFactory</b>	<code>newInstance()</code> Returns Android's implementation of <code>SAXParserFactory</code> .
<b>abstract SAXParser</b>	<code>newSAXParser()</code> Creates a new instance of a <code>SAXParser</code> using the currently configured factory parameters.

## AsyncTask 類別

`public abstract class AsyncTask extends Object`

`java.lang.Object`

↳ `android.os.AsyncTask<Params, Progress, Result>`



方法:

Protected methods	
<b>abstract</b> <b>Result</b>	<code>doInBackground(Params...params)</code> Override this method to perform a computation on a background thread.
<b>void</b>	<code>onCancelled()</code> Applications should preferably override <code>onCancelled(Object)</code> .
<b>void</b>	<code>onCancelled(Result result)</code> Runs on the UI thread after <code>cancel(boolean)</code> is invoked and <code>doInBackground(Object[])</code> has finished.
<b>void</b>	<code>onPostExecute(Result result)</code> Runs on the UI thread after <code>doInBackground(Params...)</code> .
<b>void</b>	<code>onPreExecute()</code> Runs on the UI thread before <code>doInBackground(Params...)</code> .
<b>void</b>	<code>onProgressUpdate(Progress... values)</code> Runs on the UI thread after <code>publishProgress(Progress...)</code> is invoked.

## ContextCompat 類別

`public class ContextCompat extends Object`

`java.lang.Object`

↳ `android.support.v4.content.ContextCompat`

方法:

Public methods	
<b>static final</b> <b>Drawable</b>	<code>getDrawable( Context context, int id)</code> Returns a drawable object associated with a particular resource ID.

## Fragment

1. `Fragment`
2. `FragmentManager`
3. `FragmentTransaction`

**Fragment(片段)**必須一律嵌入 **Activity** 中，而主要 **Activity** 的生命週期會直接影響片段的生命週期。例如，當 **Activity** 暫停時，其中的所有片段也會一併暫停，而當 **Activity** 遭到刪除時，所有片段也會一併刪除。不過當 **Activity** 執行時(該 **Activity** 會處於繼續進行[生命週期狀態](#))，可以個別操縱所有片段，例如新增或移除片段。當進行片段交易這類操作時，也可以將片段加到 **Activity** 所管理的返回堆疊中—**Activity** 中的所有返回堆疊項目均為所發生片段交易的記錄。返回堆疊可讓使用者復原片段交易，只要按下 [返回] 按鈕即可。

在 **Android 3.0 (API 級別 11)** 中導入了片段，主要目的是為了在大型螢幕(例如平板電腦)上支援更多動態和彈性 **UI** 設計。由於平板電腦的螢幕比手機大上許多，因此有更多空間可結合及交換 **UI** 元件。片段可實現這種介面設計，而不必管理複雜的檢視階層變更。將 **Activity** 的版面配置劃分成片段後，就可以修改 **Activity** 在執行階段的外觀，以及保留 **Activity** 所管理返回堆疊的相關變更。

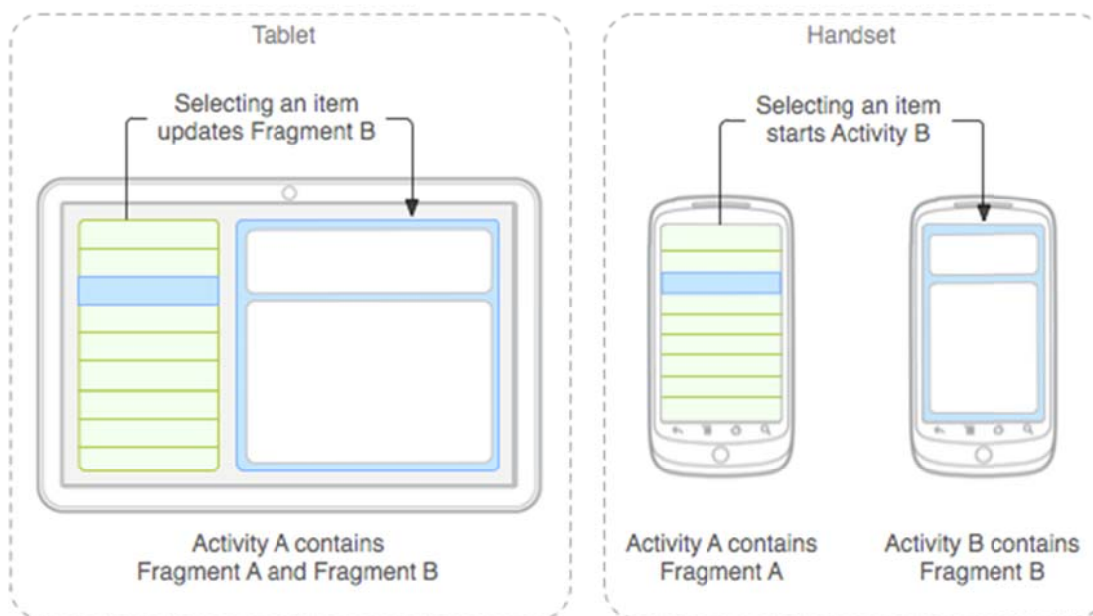


圖. 片段所定義的兩個 **UI** 模組如何針對平板電腦設計合併成單一 **Activity**

一般來說，至少必須實作下列生命週期方法：

### [onCreate\(\)](#)

系統會在建立片段時呼叫這個方法。在實作這個方法時，必須初始化您想保留的必要片段元件，以便恢復已暫停或停止的片段。

### [onCreateView\(\)](#)

系統會在片段初次顯示其使用者介面時呼叫這個方法。必須透過這個方法傳回 **View** (片段版面配置的根目錄)，才能顯示片段的 **UI**。如果片段並未提供 **UI** 的話，則可以傳回空值。

## onPause()

系統會在使用者初次離開片段時呼叫這個方法（即使使用者這麼做未必會刪除片段）。通常需要透過這個方法提交要在目前的使用者工作階段以外保留的任何變更（原因在於使用者可能不會返回）。

註:大多數應用程式都至少必須針對每個片段實作這三個方法

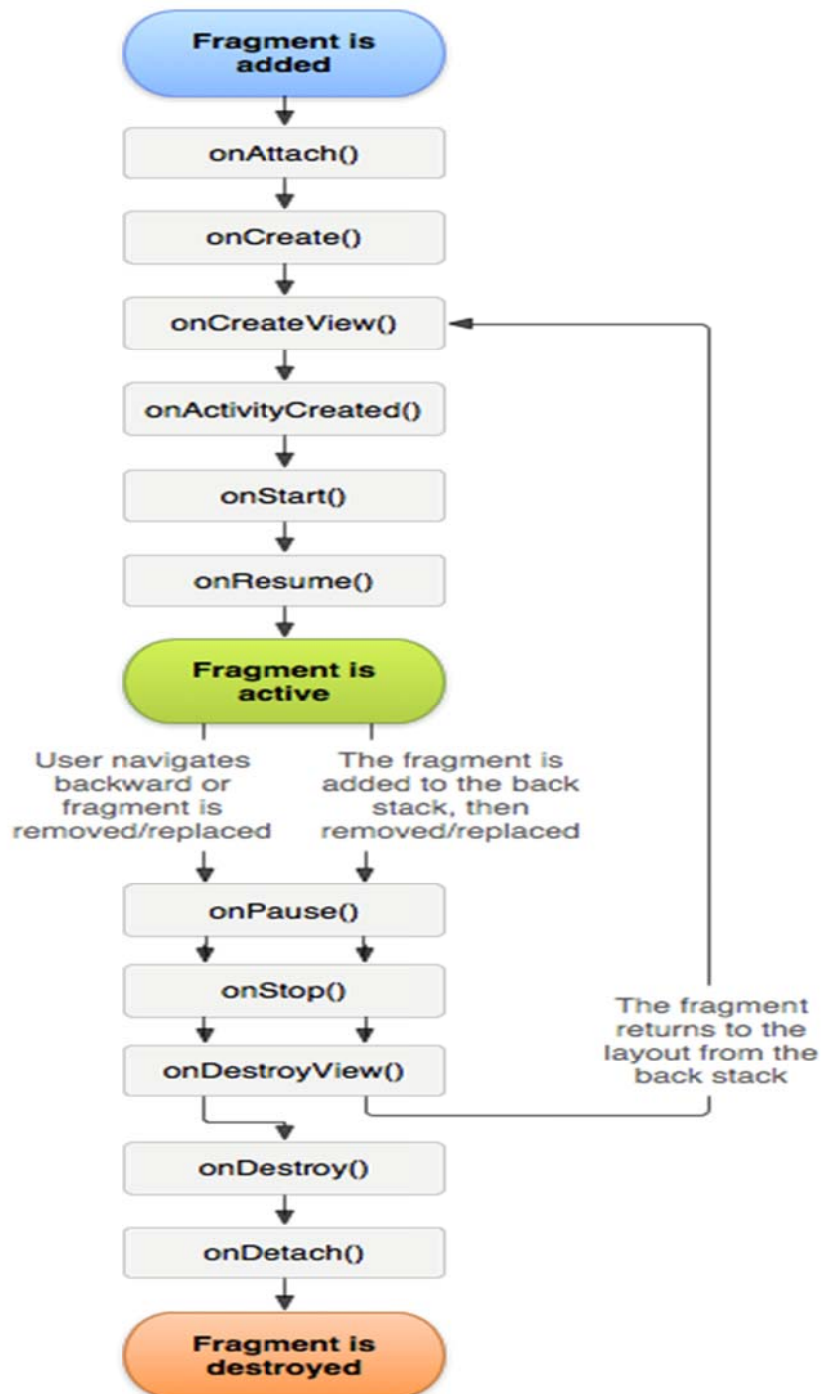


圖.片段的生命週期（當其中的 Activity 處於執行狀態時）

例.

```
public static class ExampleFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
                             container, Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.example_fragment, container,
                               false);
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <fragment android:name="com.example.news.ArticleListFragment"
        android:id="@+id/list"
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="match_parent" />
    <fragment android:name="com.example.news.ArticleReaderFragment"
        android:id="@+id/viewer"
        android:layout_weight="2"
        android:layout_width="0dp"
        android:layout_height="match_parent" />
</LinearLayout>
```

```
FragmentManager fragmentManager = getFragmentManager()
FragmentManager fragmentManager =
    fragmentManager.beginTransaction();
```

```
ExampleFragment fragment = new ExampleFragment();
fragmentTransaction.add(R.id.fragment_container, fragment);
fragmentTransaction.commit();
```